THINK C

The Professional's Choice

User's Guide

Credits

Documentation David Allcott, Bob Foster, Bonnie Hill, Jeff Mattson, Jeanne Munson, Susan Rona, and Cambridge Publications

Development Patrick Beard, Walter Bright, Thomas Emerson, Bob Foster, Greg Howe, Michael Kahl, Darrell LeBlanc, John Micco, Pat Nelson, Daniel Podwall, and Phil Shapiro

Quality Assurance Celso Barriga, Constantine Hantzopoulos, Kevin Irlen, Yuen Li, and Christopher Prinos

Technical Support Celso Barriga, Colen Garoutte-Carson, Rick Hartmann, and Scott Shurr

Project Management David Allcott, Constantine Hantzopoulos, and David Neal

Product Management Steve Levine and Peggy Liu

Copyright © 1989, 1993, 1994 Symantec Corporation.
All Rights Reserved. Printed in U.S.A.

Symantec Corporation
10201 Torre Avenue
Cupertino, CA 95014
(408) 253-9600

Symantec C++, THINK C, THINK Reference, and THINK Pascal are trademarks of Symantec Corporation. Other brands and their products are trademarks of their respective holders and should be noted as such.

The *THINK C User's Guide* is copyrighted and all rights are reserved. Information in this document is subject to change without notice and does not represent a commitment on the part of Symantec Corporation. The software described in this document is furnished under a license agreement. The document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Symantec Corporation. The *THINK C User's Guide* contains samples of names and addresses to illustrate features and capabilities of THINK and Symantec C++. Any similarities to names and addresses of actual individuals is purely coincidental.

SYMANTEC CORPORATION MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE ENCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY, OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WHICH VARY FROM STATE TO STATE.

SYMANTEC LICENSE AND WARRANTY

ıich accompanies this license (the "Soft-
›perty of Symantec or its licensors and is
›yright law. While Symantec continues to
e, you will have certain rights to use the
›ur acceptance of this license. Except as
l by a license addendum which accompa-
your rights and obligations with respect to
›ftware are as follows:

copy of the Software on a single comput-

ne copy of the Software for archival pur-
he software onto the hard disk of your
tain the original for archival purposes;

Software on a network, provided that you
copy of the Software for each computer
he Software over that network;

ritten notice to Symantec, transfer the Soft-
anent basis to another person or entity,
u retain no copies of the Software and the
s to the terms of this agreement; and

le person uses the computer on which the
lled at least 80% of the time, then after re-
›leted product registration card which ac-
oftware, that person may also use the
ngle home computer.

e documentation which accompanies the

ıse, rent or lease any portion of the Soft-

engineer, decompile, disassemble, modi-
:e any attempt to discover the source code
or create derivative works from the Soft-

revious version of the Software in addition
version received as a replacement of the
pon upgrading the Software, all copies of
must be destroyed except for copies re-
al purposes.

ıey Back Guarantee:

ginal licensee of this copy of the Software
ed with it for any reason, you may return
›duct, together with your receipt, to Sy-
:horized dealer, postage prepaid, for a full
ıe during the sixty day period following
ou of the Software.

nty:

ıts that the media on which the Software is
od or refund the money you paid for the S
Symantec does not warrant that the Software
requirements or that operation of the Softw
interrupted or that the Software will be erro

THE ABOVE WARRANTY IS EXCLUSIV
OF ALL OTHER WARRANTIES, WHETHER E
PLIED, INCLUDING THE IMPLIED WARRA
CHANTABILITY, FITNESS FOR A PARTICU
AND NONINFRINGEMENT. THIS WARRAN
SPECIFIC LEGAL RIGHTS. YOU MAY HAVE
RIGHTS, WHICH VARY FROM STATE TO S

• Disclaimer of Damages:

REGARDLESS OF WHETHER ANY REMEDY
HEREIN FAILS OF ITS ESSENTIAL PURPOSE
WILL SYMANTEC BE LIABLE TO YOU FOR
CONSEQUENTIAL, INDIRECT OR SIMILAR
CLUDING ANY LOST PROFITS OR LOST D
OUT OF THE USE OR INABILITY TO USE T
EVEN IF SYMANTEC HAS BEEN ADVISED
BILITY OF SUCH DAMAGES.

SOME STATES DO NOT ALLOW THE LIMI
CLUSION OF LIABILITY FOR INCIDENTAL
QUENTIAL DAMAGES SO THE ABOVE LIM
EXCLUSION MAY NOT APPLY TO YOU.

IN NO CASE SHALL SYMANTEC'S LIABILIT
PURCHASE PRICE FOR THE SOFTWARE. T
and limitations set forth above will apply re
whether you accept the Software.

• U.S. Government Restricted Rights:

RESTRICTED RIGHTS LEGEND. Use, duplic
sure by the Government is subject to restrict
in subparagraph (c) (1) (ii) of the Rights in
and Computer Software clause at DFARS 25
subparagraphs (c) (1) and (2) of the Comme
Software-Restricted Rights clause at 48 CFR 5
plicable, Symantec Corporation, 10201 Torr
pertino, CA 95014.

• General:

This Agreement will be governed by the law
California. This Agreement may only be mo
cense addendum which accompanies this l
written document which has been signed b
Symantec. Should you have any questions
Agreement, or if you desire to contact Syma
son, please write: Symantec Customer Sales
10201 Torre Avenue, Cupertino, CA 95014.

Language Software

If the Software is a Symantec language pro
have a royalty-free right to include object co
the libraries in programs that you develop
ware and you also have the right to use, di

this page intentionally left blank for compuserv ad copy page 1 of 4

this page intentionally left blank for
compuserv ad copy page 2 of four

this page intentionally left blank for
compuserv ad copy page three of four

this page intentioanlly left blank for compuserv ad copy page four of four

Eddy Award Winner** for **Best New Developer Tool
*– MacUser Editors Choice Awards, **1993***

*"A distinct improvement over ResEdit....
The tool of choice for serious Mac developers."*
– MacTech / MacTutor

"Resorcerer's data template system is amazing!"
– Bill Goodman, author of Compact Pro

"Nuke ResEdit! Resorcerer is mission-critical for us."
– Dave Winer, Userland Frontier

"Every Macintosh developer should own a copy of Resorcerer."
– Leonard Rosenthol, Aladdin Systems

*"Resorcerer will pay for itself many times over in saved time and effort.
... a wealth of time-saving tools. No developer should be without it."*
– MacUser review, Dec. 1992

"The color pixel editors are wonderful! A work of art!"
– Dave Winzler, author of Microseeds Redux

"The template that disassembles 'PICT's is awesome!"
– Bill Steinberg, author of Pyro! and PBTools

"Resorcerer proved indispensible in its own creation!"
– Doug McKenna, author of Resorcerer

MACUSER EDITORS' CHOICE AWARD · 1993 · WINNER ·

Only $177
With Coupon Below

Downloadable Demos/Updaters:
AppleLink: Software Sampler
AOL: Software Libs/Development
CompuServe: MACDEV/Tools

Version 1.2.4

RESORCERER®

The Premiere Macintosh Resource Editor

MATHEMÆSTHETICS, INC. • P.O. Box 67156 • Chestnut Hill • MA • 02167 • USA
Phone: (800 or 617) 738-8803 • AmericaOnline/AppleLink: Resorcerer • Internet: resorcerer@aol.com

**Yes, I want to save 30% off the usual Resorcerer price of $256. Price per copy with this coupon is $177.
For US/Canada shipping, add $5/copy. Int'l AirMail Europe, add $20/copy; Japan/Pacific, add $30/copy.**

Name _____
Company _____
Address _____
_____
_____
_____
Phone _____
E-mail _____

Offer good through 12/31/94

◯ **Check enclosed (US funds only)**
◯ **Domestic Purchase Order Attached**
◯ **Visa / Mastercard (with signature):**

☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ Exp. ☐☐ / ☐☐

THC 3/94

Contents

Part One: Getting Started

1 Welcome 3
- If you don't read manuals 3
- If you're an experienced THINK C user 3
- If you're new to THINK C or the C language 3

The THINK Development Environment. 5
What You Need 6
- Which Macintosh models? 6
- How much disk space? 6
- Which System/Finder? 6

What's in the Package 6
What's in the User's Guide 6
- Getting Started 7
- Learning the THINK Environment 7
- Using THINK Project Manager 7
- Using the THINK C Compiler 9
- Using Object Extensions 11
- Advanced Topics 11
- Reference 11
- Appendixes 12

Conventions 12
What You Should Know 13
- Learning C 13
- Learning to write Macintosh programs 15
- Apple Computer, Inc. 17
- Apple Programmer's and Developer's Association (APDA) . . 18
- CompuServe 18

2 Installation 19
Installation Overview 21
Before You Install 21
Installing Symantec C++ 22
Installing Just Some of the Product 24
Installing THINK C 26
Installing Just Some of the Product 29
Installing AppleScript Runtime™ 30
Configuring the THINK Reference Databases 31

Part Two: Learning the THINK Environment

3 Tutorial: Hello World 35
Creating the Project 37
Creating the Source File 41
Compiling the Source File 42
Did you get an error? 43
Adding the Library 44
Running the Project 46
Creating the Application 48
Where to Go Next 49

4 Tutorial: MiniEdit 51
Creating the Project 53
Adding the Source Files 56
Compiling and Running the Project 59
Fixing a Bug 59
Running the Project Again 62
Building the Application 64
Using a Resource File 66
Finishing Up 66
Where to Go Next 66

5 Tutorial: Bullseye 69
Opening the Bullseye Project 71
Turning the Debugger On 71
Generating the debugging tables 72
Running the project 72
Watching the Program Run 72
The Source window 73
Stepping through statements 74
Stepping into functions 75
Stepping out of functions 75
Tracing every statement 76
Setting a breakpoint 77
Letting the program run 79
Stopping the program 79
Viewing other files 79
Examining and Setting Variables 80
The data window 81
Examining variables 81
Changing the value of a variable 85
Examining structs, classes, and arrays 86
Expressions and Contexts 90
How and when the source debugger evaluates expressions . 91
Display formats 91
Quitting the Debugger 93

6 Tutorial: LearnOOP 95
What Does LearnOOP Do? 97
How Does LearnOOP Work? 99
How Do Objects Know How to Draw Themselves? 100
The LearnOOP Class Declarations 104
How Do You Create an Object? 107
How Do You Define a New Class? 108
But How Does It Know? 110
Where to Go Next 111

Part Three: Using the THINK Project Manager

7 Overview 115
The THINK Project Manager 117
The Project 117
Writing a Program 117
Creating source files 118
Adding libraries 118
Compiling the program 118
Running the program 118
Debugging the program 119
Building the application 119
Using the THINK Project Manager 119

8 Organizing Projects 121
What Is a Project? 123
The Project Window 123
What You Can Put in Your Project 126
Source files 127
Libraries 127
Resources 128
Notes 129
Creating a New Project 129
Using project models 129
Using project stationery pads 131
Creating your own project models 133
Opening an Existing Project 133
Closing Projects 135
Adding Files to Projects 135
Adding a file in the THINK Editor 135
Adding a file on disk 136
Selecting Files 137
Removing Files from Projects 137
Setting the Project Type 137
Segmenting Projects 139
Letting the THINK Project Manager segment your project . 139
Moving files into segments 140
Moving entire segments 141
Deleting a segment 141
Changing a segment's name and attributes 141

Compressing Projects 142
Recovering Corrupted Projects 142
Setting Project Options. 143
Using the Options dialog 144
Changing File Extensions 146
Changing the translator for an extension 147
Adding an extension 147
Deleting an extension 148
Changing Your Project Window View 148
Viewing your project's components 148
Viewing segment numbers 150
Project Limits 151

9 Organizing Files and Folders 153
Organizing Your Files 155
Don't put project folders in the THINK Tree 155
Avoid duplicate file names in trees 156
Don't move the THINK Project Manager application . . . 156
Moving Files on Disk 156
Moving a source file 156
Moving a library 157
Moving other files 157
Moving files to another machine 157
A note about search times 157
Managing Files 158
Using shielded folders and project-specific folders 158
Using aliases. 158
Placing files into folders 160

10 Editing Files 163
Creating and Opening Files 165
Creating a new file 165
Opening a text file 165
Opening text files automatically 166
Opening a source file. 166
Opening header files 166
Opening the current selection 167
Editing a File 168
Typing text 168
Undoing changes to a file 168
Selecting lines 168
Indenting 169
Shifting blocks right and left 169
Balancing parentheses, brackets, and braces 169
Changing font and tab settings 170
Moving Around a File 170
Using the arrow and function keys 171
Scrolling to the insertion point. 172
Moving to a specific line. 172

Looking Up Definitions and Documentation 173
Looking up function, class, and global variable definitions . 173
Looking up method definitions 173
Looking up ANSI and Macintosh Toolbox documentation . 174
Printing Files and Windows 174
Selecting and Arranging Windows 175
Closing and Saving Files 176
Closing a file 176
Saving a file 176
Saving a file with a different name 176
Saving and closing all open files 177
Saving files automatically 177
Executing MPW Commands 177
Using Markers 179
Adding markers 179
Going to a marker 179
Deleting a marker 181
Using Your Own Editor 181
Using a THINK-compatible editor 181
Using another editor 182
Searching and Replacing. 182
Finding a string 183
Search options 184
Replacing a string 185
Setting things up for searching later 185
Finding nonprinting characters 186
Searching through multiple files. 186
Disabling multi-file search. 190
Searching for a Pattern (Grep). 190
Patterns 190
Simple patterns 190
Complex Patterns 191
Sub-patterns 192
Constraining patterns 193
Replacing with Grep 193
Grep Examples 194

11 Running Programs 197

Running a Program 199
A word of caution 201
When Something Goes Wrong 201
The bug alert 201
The Compile Errors window 202
The Link Errors window 205
Compiling Source Files 206
Compiling files not in the project 206
Compiling files already in the project 206
Checking files without compiling 206
Compiling files that auto-make misses. 206

Examining What the Compiler Does 209
Viewing the preprocessor output 209
Disassembling your code 210
Generating a link map 210
Precompiling Headers 212
Creating your own precompiled headers 213

12 Debugging Programs 215
Running with the Debugger 217
Turning the debugger on 217
Running the project 218
The Debugger Windows 219
The Source window 219
The Data window 220
Working with the Source Window 221
The current statement and the selected statement 222
The current function indicator 223
Viewing other source files in the Source window 224
Viewing header files in the Source window 224
Editing files while debugging 226
Searching in the Source window 227
Setting Breakpoints 228
Simple breakpoints 229
Setting breakpoints in another file 230
Setting temporary breakpoints 230
Setting conditional breakpoints 231
Using Debugger() and DebugStr() 233
Controlling Execution 233
Go 234
Trace 234
Step 234
Step In. 234
Step Out 234
Stop 235
Go Until Here 235
Skip To Here 235
Stepping continuously 236
Working with the Data Window 236
Entering an expression 237
Editing expressions 238
Removing expressions 238
Setting values 238
Formatting values 238
Displaying and changing contexts 240
Evaluating expressions 240
Calling functions from the Data window 241
Working with expressions 241
Examining structs and arrays 242
Error Messages in the Debugger 243
Saving the Debugger State 246

Debugging Optimized Code 246
For both THINK C and Symantec C++. 246
For THINK C 247
For Symantec C++ 247
Debugging Options 247
Use Source Debugger 248
Use second screen 248
Update program windows. 248
Always save session. 248
Allow dataview function calls 248
Store debug info separately 249
Using Low-Level Debuggers 249
Using the Monitor command with TMON Pro 249
Using the Monitor command with other debuggers . . . 250
Leaving the low-level debugger. 250
Quitting the Debugger 250
Memory Considerations 250

13 Building Projects 253
Putting a Project Together 255
Building Applications. 256
Setting the application file type and creator. 257
Using a larger jump table with Far CODE 257
Using more global data with Far DATA 258
How Far CODE and Far DATA work 258
Mixing Near CODE and DATA with Far CODE and DATA . 258
Using a separate STRS component 259
Setting the partition size and SIZE resource flags 259
Building Desk Accessories and Device Drivers 262
Setting the project type. 262
How drivers work 264
How to write main( ) for a driver 265
Getting the event record pointer from paramBlock . . . 265
Global data in drivers 266
Using driver globals in callback and trap intercept routines 267
Using libraries in drivers 268
Setting the fields of a driver's header 268
Opening an open driver 271
How to return from a driver 271
The jIODone problem 272
Multi-Segment drivers 273
Building Code Resources 274
Setting the project type. 275
How to write main() for a code resource. 276
Global data in code resources 277
Using libraries in code resources 278
Locking code resources 279
Code resource headers. 281
Merging code resources into files 283
Multi-segment code resources 283

Building Libraries 284
Projects as libraries 285
Binary libraries 285

14 Browsing Classes. 287
Using the Class Browser 289
Finding the Declaration of a Class 289
Finding a class declaration with the Class Browser 289
Finding a class declaration with the editor. 290
Finding the Definition of a Method 290
Finding a method definition with the Class Browser . . . 291
Finding a method definition with the editor 291
Keyboard Shortcuts in the Class Browser 293
Class Browser Summary 294

Part Four: Using the THINK C Compiler

15 Writing C Code 297
How THINK C Implements C 299
Identifier length and capitalization 299
How THINK C looks for header files 299
Once-only headers 299
Using register variables 300
Integer Representation. 302
Integer limits 303
Floating-Point Representation 305
Floating-point parameters and limits. 307
THINK C Extensions 310
Disabling trigraphs 312
enums of any size 312
Inline assembly. 313
pascal keyword. 313
C++ style comments 313
short double type 313
Comments after #else and #endif 313
Inline function definitions 313
Low-memory global definitions 314
MC68881 unary inline functions 314
Function prototypes with (...) 314
Dimensionless arrays 315
void * with function pointers 315
THINK_C predefined symbol 315
Using #pragma Directives. 315
The #pragma parameter directive. 315
The #pragma nooptimize directive 316

Accessing Option Settings in Your Code 317
Internal compiler options 318
Set Project Type options 318
Type options 319
Language extension options 319
Enumerated type option 320
Pascal string option 321
Array alignment option. 321
Object-oriented programming options. 322
Type-checking options 322
Generating processor-specific code options 323
Debugging options 324
Global optimizer options 324
Other optimization options 326

16 Compiling C Code 327
Compiling ANSI-Conformant Code 329
Compiling C++-Compatible Code. 330
Type Checking 331
Checking pointer types. 332
Enforcing prototype use 333
Compiling Processor-Specific Code 335
Compiling code for the MC68020 335
Compiling code for the MC68881 336
Compiling Code to Debug It 336
Always generate stack frames 337
Generate profiler calls 338
MacsBug Names 338
Using the THINK C Project Prefix 338

17 Optimizing Code 341
What an Optimizer Is. 343
Why use an optimizer 343
When not to use an optimizer 343
Using the Global Optimizer 344
Induction variable elimination 345
CSE elimination 345
Code motion 346
Register coloring 346
Using Local Optimizations 347
Defer & combine stack adjusts 347
Suppress redundant loads 348

18 Calling Toolbox Routines 349
Calling Toolbox Routines 351
Passing arguments to Toolbox routines 352
Working with Pascal strings 353
Calling AppleTalk routines 354
Using MacHeaders 355
Editing the MacHeaders file 356

The Macintosh Header Files 357
The Mac #includes folder 358
The Apple #includes folder 358
The THINK #includes folder 359
Working with Pascal routines 362
Pascal callback routines 362
Calling Pascal routines indirectly 363
Working with Floating-Point Formats 364
Working with SANE 364
The extended type 365
A note on SANE implementations 366

19 Porting Code 367
Porting Code to THINK C. 369
Changing the size of doubles 369
Changing the alignment of char arrays in structs. 370
Porting code from other versions of THINK C 370
Porting Code to Symantec C++ 370
Calling C Routines 371
C calling sequence 372
C function entry 372
C function exit 373
Functions that return struct, union, or double 373
Functions that accept a variable number of arguments . . 374
Calling Pascal Routines 375
Pascal calling sequence 376
Pascal function entry 376
Pascal function exit 377
Calling C++ Routines 377
Finding the mangled name for a C++ function 378

20 Writing Inline Assembly 379
Using the Inline Assembler 381
Writing assembly language code 381
Writing processor-specific assembly code 383
Using C identifiers in assembly language 384
Using floating-point literals 386
Labels and branching. 386
Returning from a function 389
Multiple entry points 389
Using the Macintosh Toolbox in assembly language . . . 389
Register usage 393
Differences from Other Assemblers 393
Hexadecimal constants 393
Directives. 394
Addresses. 395
Tips 395
Using constants. 396
Local storage 396
Instruction size 396
Multi-line assembly language macros 397

21 Using the Standard Libraries 399
What Are the Standard Function Libraries? 401
Special versions of the standard libraries 402
Customizing the ANSI library 403
Using the Online Standard Libraries Reference 404
Looking up a function 405
Looking up a topic 406
Moving around THINK Reference 406
Reading a function reference page 407
Using the tables of contents 408
Finding lost databases 410

Part Five: Using Object Extensions

22 Object-Oriented Programming 415
Contents 415
Overview 417
Objects and Messages 417
Classes 418
Inheritance and Polymorphism 419
Objects and the Macintosh Interface 420
Working with Objects 421
What classes should I define? 421
When should I create a subclass? 421
What should be a method? 421
When should I use procedural programming? 422
Where to Go Next 422

23 Using Objects in THINK C 423
Introduction 425
What you should know 425
A note on terminology 425
Using the Object Extensions 426
Overview 426
Declaring a Class 428
Declaring root classes 429
Declaring subclasses 430
Declaring and Using Objects 430
Creating and deleting objects 430
Assignment and objects 432
Objects and memory 433
Accessing Members of a Class 433
Declaring Members of a Class 435
Declaring instance variables 435
Declaring class variables 437
Declaring methods 438
Virtual and nonvirtual methods 438
Class methods 439
Examples of methods 439
Declaring circular and forward references 440

Defining and Using Methods 441
Access Control 443
Using access specifiers 444
Access specifiers in subclass declarations 445
Protected access 446
Constructors and Destructors 448
Declaring and defining constructors 448
How constructors are invoked 449
Using access specifiers with constructors 450
Destructors 450
Allocators and Deallocators 451
Allocators 452
Deallocators 453
Default allocator and deallocator 454
Tips and Techniques 454
Organizing your classes 455
Objects and the Macintosh 455
Direct and indirect classes 458
Object Extension Options 458
Turning on the object extensions 458
Classes are indirect by default 459
Methods are virtual by default 460
Generate class names 460
Optimize monomorphic methods 460
Special Functions 460
Differences Between THINK C and C++ 461
General differences 461
Keywords 462
Value objects and reference objects 462
THINK C extensions 463
Access declarations, friends, and multiple inheritance . . . 463
Constructors and destructors 463
Scope resolution operator 463
Sizeof operator not allowed 463

Part Six: Advanced Topics

24 Using SourceServer 467
About SourceServer 469
Overview 469
Features 471
Limitations 472
Using SourceServer: A Tutorial 473
Creating and mounting a new project 473
Checking in a revision 474
Opening files in a Projector database 475
Checking out files 475
Modifying read-only files 476
Checking in your revisions 477

The Check In Dialog 478
Project pop-up menu 478
User field 478
Select files area 479
Task and comment fields 479
Keep read-only, Keep modifiable, and Delete my copy buttons 479
Touch mod date check box 479
Verify check box 480
Check In button 480
Done button 480
The Check Out Dialog 480
Select files area 481
By Name field 483
User field 483
Done button 483
Check Out button 483
Undo Checkout button 483
Info button 484
Checkout To directory field 484
Task and Comment fields 484
Read-only and Modifiable buttons 485
Branch check box 485
Touch mod date 485
Verify check box 485
The Name Revisions Dialog 486
Creating Branches 486
Components of a Projector Database 487
Projector database 488
Revision trees 489
Branches 491
Symbolic names 491
SourceServer Database Administration 491
Creating nested databases 492
Moving, renaming, and deleting projects 493
Deleting revisions 493
Removing a file from a project 494
Using the Command-Line Interface 494
SourceServer Command Descriptions 495

25 Profiling Projects 505
Using the Profiler 507
Setting options to profile the most functions possible . . 508
Logging Time Statistics 509
Tracing Your Functions 510
Modifying the Profiler 511
Changing where to print 511
Changing which timer to use 512
Changing the profiler's report 512
Recompiling the profiler 513

Summary 514
User functions 514
Internal functions 515

26 Using Object Files 517
Adding MPW .o Files to Your Project 519
Create the vocabulary file 519
Edit the vocabulary file 521
Create the final object code. 521
Converting Large .o Files 521
Create a list of symbols 522
Split up the symbols 522
Make the pieces 523
Use the .o Converter to convert the pieces 523
Setting .o Converter Options. 523
The use Toolbox trap list option 524
The use '.v' file option 524
Conversion Limits 525
Segmentation limits 525
Computed references. 525
Reference records 525
SADE records 525
MPW run-time routines 525

Part Seven: Reference

27 Using THINK Rez 529
The Resource Compiler 531
Using a resource compiler 531
Standard type declaration files. 532
Structure of a Resource Description File 532
Sample resource description file 533
Resource Description Statements 534
Syntax notation. 534
Special terms 535
Data—specify raw data 535
Type—declare resource type 536
Resource—specify resource data 547
Labels 551
Built-in functions to access resource data 551
Declaring labels within arrays 552
Label limitations 553
Using labels: two examples. 554
Preprocessor Directives 557
Variable definitions 558
Header file processing 559
If-then-else processing 560

Resource Description Syntax 561
Numbers and literals 561
Expressions 562
Variables and functions 563
Strings 565
Setting THINK Rez Options. 568
Resource alignment 568
Redeclared types ok. 568
Prefix String 569
Differences from MPW Rez 569

28
Using SADeRez 571

What Is SADeRez? 573
Running SADeRez 573
Choosing Input Files 574
Choosing the resource file. 575
Choosing a description file 575
Choosing #INCLUDE paths 576
Choosing Output Files 578
Setting Options 579
Choosing the types to decompile 580
Preprocessor 581
Saving and Restoring Options 581
The Messages Window 582

29 Language Reference 585

Introduction 587
Implementation-defined behavior 587
Undefined behavior 587
Setting ANSI conformance. 587
About the standard libraries 587

Language Reference 588
2.1.1.2 Translation phases 588
2.1.1.3 Diagnostics. 588
2.1.2.2.1 Program startup 588
2.1.2.3 Program execution 588
2.2.1 Character sets 588
2.2.1.2 Multibyte characters. 588
2.2.4.2.1 Sizes of integral types <limits.h> 588
3.1.2 Identifiers. 589
3.1.2.2 Linkages of identifiers 589
3.1.2.5 Types 589
3.1.3.4 Character constants 591
3.1.7 Header names 592
3.2.1.2 Signed and unsigned integers 592
3.2.1.3 Floating and integral 592
3.2.1.4 Floating types. 592
3.3 Expressions. 592
3.3.2.3 Structure and union members 592
3.3.3.4 The sizeof operator 593
3.3.4 Cast operators 593
3.3.5 Multiplicative operators 593
3.3.6 Additive operators 593
3.3.7 Bitwise shift operators 593
3.3.8 Relational operators 593
3.5.1 Storage-class specifiers 594
3.5.2.1 Structure and union specifiers 594
3.5.2.2 Enumeration specifiers. 595
3.5.3 Type qualifiers 595
3.5.4 Declarators 595
3.6.4.2 The switch statement 595
3.8.1 Conditional inclusion 595
3.8.2 Source file inclusion 596
3.8.3 Macro replacement. 596
3.8.6 Pragma directives 596
3.8.8 Predefined macro names. 599
4.1.5 Common definitions <stddef.h> 600
4.2 Diagnostics <assert.h> 600
4.3.1 Character-testing functions 600
4.5.1 Treatment of error conditions 600
4.5.6.4 The fmod function 600
4.7.1.1 The signal function 601
4.9.2 Streams 601
4.9.3 Files. 601
4.9.4.1 The remove function 601
4.9.4.2 The rename function 601
4.9.5.2 The fflush function 601
4.9.6.1 The fprintf function 602
4.9.6.2 The fscanf function 602
4.9.9.1 The fgetpos function 602
4.9.9.4 The ftell function. 602
4.9.10.4 The perror function 602
4.10.3 Memory management functions 602

4.10.4.1 The abort function . . . 602
4.10.4.3 The exit function . . . 602
4.10.4.4 The getenv function . . . 602
4.10.4.5 The system function . . . 603
4.11.6.2 The strerror function . . . 603
4.12.1 Components of time . . . 603
4.12.2.1 The clock function . . . 603
THINK C Extensions . . . 603
Inline assembly . . . 603
pascal keyword . . . 603
C++ style comments . . . 603
short double type . . . 603
Identifiers after #else and #endif . . . 604
Inline function definitions . . . 604
Low memory global definitions . . . 604
MC68881 unary inline functions . . . 604
Function prototypes . . . 604
Dimensionless arrays allowed . . . 605
void * . . . 605
Predefined symbols . . . 605
THINK C Object Extensions . . . 605
Keywords . . . 605

30 THINK Project Manager Menus . . . 607
The Apple Menu . . . 611
About THINK Project Manager . . . 611
The File Menu . . . 611
New/ New Project . . . 611
Open/ Open Project . . . 613
Open Selection . . . 614
Close . . . 614
Save . . . 614
Save As . . . 614
Save a Copy As . . . 615
Revert . . . 615
Page Setup . . . 615
Print . . . 615
Modify Read-Only . . . 615
Quit . . . 615

The Edit Menu 616
Undo 616
Cut 616
Copy 617
Paste 617
Clear 617
Select All 617
Options 617
THINK Project Manager options 619
Preferences page 620
Editor page 622
Debugging page 624
Extensions page 626
Project Window page. 627
.o Converter options 628
THINK C options 628
Language Settings Page 629
Compiler Settings page 633
Code Optimization page. 639
Debugging page 642
Prefix page 644
THINK Rez options 645
Set tabs & font 646
Shift Left 647
Shift Right 647
Balance 647
The Search Menu 648
Find 648
Enter Selection 650
Find Again 650
Replace 650
Replace & Find Again 651
Replace All 651
Find in Next File 651
Find in THINK Reference. 652
Go To Line 652
Mark 653
Remove Marker. 654
Go To Next Error 654
Go To Previous Error. 654

The Project Menu 655
Close Project 655
Close & Compact. 655
Switch to Project 655
Set Project Type 655
Application dialog 656
Desk Accessory and Device Driver dialogs 658
Code Resource dialog 659
Remove Objects 660
Bring Up To Date 660
Check Link 660
Build Library 661
Build Application
Build Desk Accessory
Build Device Driver
Build Code Resource 661
Use Debugger 662
Run 662
The Source Menu 663
Add Window 663
Add Files 664
Remove 664
Get Info 665
Debug 665
SourceServer 666
Check Syntax 667
Preprocess 667
Disassemble 667
Precompile 667
Compile
Load Library 667
Make 668
Browser 670
The Windows Menu 671
Arrange 671
Zoom 672
Full Titles 672
Close All 672
Save All 672
Project window 672
Edit Windows 672

31 Debugger

Menus675

The Apple Menu 677
Shortcuts 677
The File Menu 677
Save All 677
Close 677
Close All 677

The Edit Menu 678
Undo 678
Cut 678
Copy 678
Paste 678
Clear 678
Copy To Data 678
The Debug Menu 679
Go 679
Step 679
Step In. 679
Step Out 680
Trace 680
Stop 680
Go Until Here 680
Skip To Here 680
Inspector 681
Monitor 681
ExitToShell 681
The Source Menu 682
Clear Breakpoint
Set Breakpoint 682
Clear All Breakpoints 682
Attach Condition 682
Show Condition 682
Edit ‘filename’ 682
The Data Menu 683
Clear All Expressions 683
Set Context 683
Show Context 683
Signed Decimal
Unsigned Decimal
Hexadecimal
Character
Pointer
Address
C String
Pascal String
Floating Point 684
Locked 685
Context Free. 685
The Windows Menu 685
projectname 685
filename.c 685
Data 685

Part Eight: Appendixes

A What's New 689
Upgrading THINK C 691
Building C++ Applications Is Simpler 691
Creating New Projects Is Easier 691
Large Projects Are Segmented for You 692
More Arrangement Options for Windows 693
Find Dialog Remembers Old Search Strings 694
More Movable Dialogs 694
Avoid Open Project Dialog 694
More Window Positions Saved 694
Project Window Is Colorized 694
Preferences Stored in Preferences File 695
Scripting Improved 695
SourceServer Improved 695

B Troubleshooting 697
How to Find Your Solution 699
First Try This... 699
Check for virus infection 699
Investigate virus alerts 700
Check for INIT conflicts 700
Launching and Opening Projects 701
Low memory 702
Missing resource error 703
Compiling a Project 703
Corrupted resource file 704
Crashing While Running Your Project 704
Pointer errors 705
Stack errors 706
Getting Unexpected Function Results 706
Using your own functions 707
Using the ANSI library 707
Using Macintosh Toolbox functions 708
Using math functions 709
Using the Debugger 709
When All Else Fails... 711

C Error Messages 713
Error Messages 714

D Glossary 773

Index 789

Symantec Customer Service Plan 807